Exceptional service in the national interest

ME469: Course Objectives

Stefan P. Domino[1,2]

[1] Computational Thermal and Fluid Mechanics, Sandia National Laboratories

[2] Institute for Computational and Mathematical Engineering, Stanford

This presentation has been authored by an employee of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the presentation and is solely responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan https://www.energy.gov/downloads/doe-public-access-plan.

SAND2018-4536 PE

Course Goal: Increased Proficiency in Numerical Approaches for Computational Fluid Dynamics

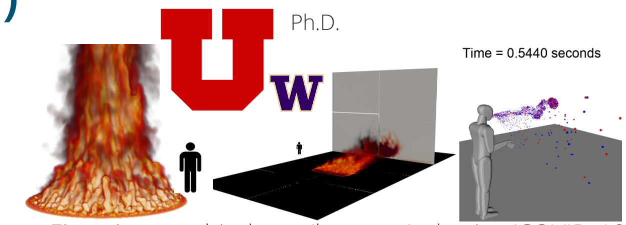
- Analysis and implementation of computational approaches applied to the simulation of fluid motion
  - In this class, we will focus on low-Mach flows in the acoustically incompressible regime
- Key ingredients
  - Approximations: from the real world to a physical/mathematical representation
  - Computations: from a continuous mathematical model to computer algorithms and, finally, to a discrete solution
- Emphasis
  - Numerical algorithms for fluid mechanics and their implementation
  - Analysis and understanding of the numerical properties and physical motivation
  - Verification and Validation (V&V)
- Computing and programming are of primary importance
- Classes are designed to work from fundamentals of discretization, to supporting complex multiphysics modeling and simulation (Mod/Sim) experience
  - Weekly lab-like sessions (Friday) will run canonical CFD cases (channels, pipes, jets, etc.)

Who Am I? Stefan P. Domino (he/him)

Distinguished Member of the Technical Staff

Adjunct Professor

Ph.D.



Fires, jets, multi-phase (buoyant) physics (COVID-19)

PHYSICS

low-Mach turbulent flows with emphasis on variable density reacting/multi-phase; buoyancy

NUMERICS

Low-dissipation, low-Mach generalized unstructured methods development for LES and DNS

VVUQ

Multi-physics code verification techniques along with structural uncertainty approaches for LES; QoI behavior for fires in crossflow

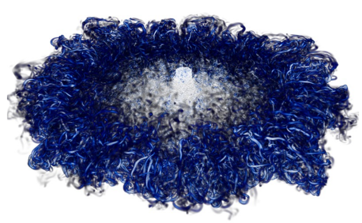
HPC

Routine massively parallel computing with unstructured, fully implicit mesh counts of O(10) billion elements on 1000-6000 nodes.

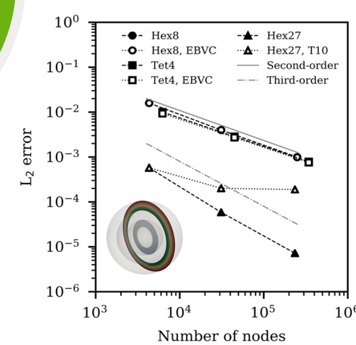
CODE

Efficient, clean coding design for modern low-Mach multi-physics applications

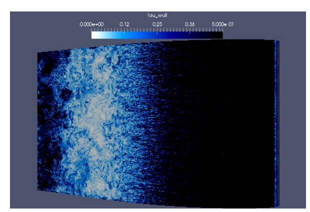
github.com/NaluCFD



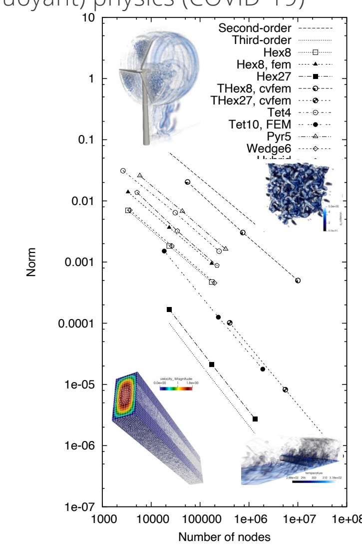
O(2) billion DNS

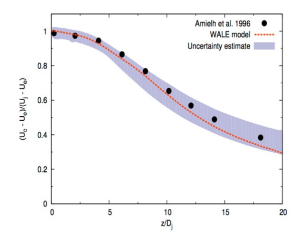


Code Verification



O(6) billion WRLES





Ven diagram and concept modified from **PresentationGO.com**

Course Support

- Stefan P. Domino - Sandia National Laboratories, spdomin@stanford.edu
- Mark Benjamin – markben@stanford.edu
- Steve Jones – Stanford HPC Center stevejones@stanford.edu

Your Turn….

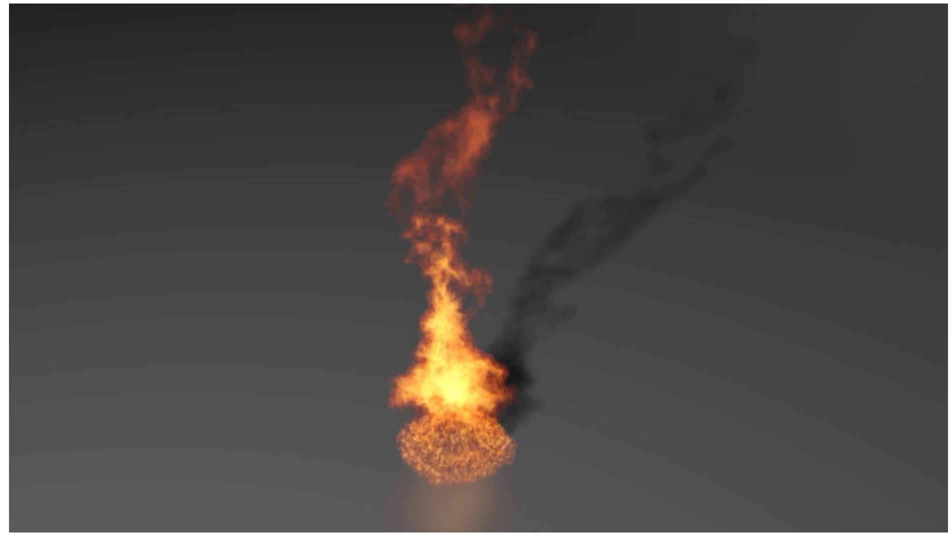
- Field of research/interest
- Relevant Experience
  - CFD, Simulation Software, etc.
  - Programming languages,
  - HPC libraries, etc.
  - Supercomputing,
  - Cluster access experience, etc.

Please provide me with a one-page overview of your name, interests, your background, and a picture

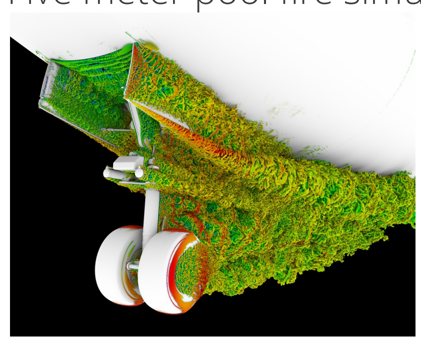
What is CFD?

- Computational Fluid Dynamics (CFD) is a branch of computer-aided science that enables numerical simulations and predictions of fluid flow behavior
- Mathematical modeling: typically a system on non-linear, coupled partial differential equations (PDEs)
- Numerical methods: Involves both discretization and solution techniques
- Software engineering: code implementation and software quality assurance (SQA)

CFD software tools are commercially available or accessible as open-source packages, however, research is ongoing to improve the methods, physical models, conceptual understanding, etc.



Five meter pool fire simulation, Domino et al., PoF (2022)



NASA simulation of a flow field around the nose landing gear of a Boeing 777

Image credit: NASA's Ames Research Center, Patrick Moran; NASA's Langley Research Center, Mehdi Khorrami; Exa Corporation, Ehab Fares

Milestones in CFD

- 1910 - Richardson: *Forecast Factory*: hand calculations with human computers, 2000 operations per week – envisioned to 60,000 humans
- 1960 - Marker and Cell methods - Harlow & Welch (Los Alamos)
- 1970 - Finite difference methods for Navier-Stokes
- 1972 - Reynolds Averaged Navier-Stokes & k-ε models (Imperial College)
- 1975 - Finite element methods for stress analysis
- 1980 - Finite volume methods (Imperial)
- 1980 - Direct Numerical Simulations of Channel Flow (Stanford)
- 1985 - Routine use in aerospace & aeronautical industries (Boeing, General Electric, ...)
- 1995 - Routine Use in "non-aero" industries (GM, Ford, ...)
- 1996 - ASCI Red (Sandia) reaches 1.3 tera-FLOPS); 10^{12} floating point operations per sec
- 2008 – Roadrunner (LANL) reaches 1.042 peta-FLOPS; 10^{15} FLOPS
- 2023 – Frontier (ORNL) is the first Exascale computer; 10^{18} FLOPS

now - widespread use in all industries!

Forecast Factory, Circa 2020

"Richardson foresaw a 'forecast factory,' where he calculated that 64,000 human 'computers,' each responsible for a small part of the globe, would be needed to keep 'pace with the weather' in order to predict weather conditions". https://celebrating200years.noaa.gov

CFD & ME469/CME369

Rich tradition of CFD at Sandia and Stanford - this class focuses on:

- Introduction to numerical methods and their properties
- Fluid mechanics flow equations and approximation levels
- Unstructured grid approaches:
  - Finite volume: Cell-centered, edge/element-based; and Finite Element Methods (FEM)
  - Hands on experience with edge/element-based and FEM solver
- Numerical methods for low-Mach/incompressible flow
- Verification and Validation (V&V) methodologies including Uncertainty Quantification (UQ)
- Extensions include:
  - Variable-density flows
  - High performance computing
  - Volume of Fluid
  - Move movement including sliding mesh and overset
  - Multiphysics couplings
  - ....others

Course Material

In-class Supplied Material

- Slides and tutorials will be posted to Canvas (generally, post-lecture)
- A new course reader, in partnership with Professor Gianluca Iaccarino, will serve as a supplement to the lecture notes and will be posted to Canvas
- Nalu: Open source software documentation
  - https://nalu.readthedocs.io/en/latest/source/theory/index.html

Optional Textbook(s)

- J. Ferziger, M. Peric: *Computational Methods for Fluid Dynamic*, Springer
- H. Versteeg, W. Malalasekra: *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, Prentice Hall
- C. Hirsch: *Numerical Computations of Internal and External Flows*, Butterworth-Heinemann
- R. Moser: *Numerical Methods in Turbulence Simulation*, Elsevier
  - Chapter 7, “Unstructured finite volume approaches for turbulence”, Domino, will be distributed in class

Computational Simulations Tools Exercised

We plan on using the open-source Nalu code base that has already been installed on the dedicated mini-cluster: me469-cluster.stanford.edu

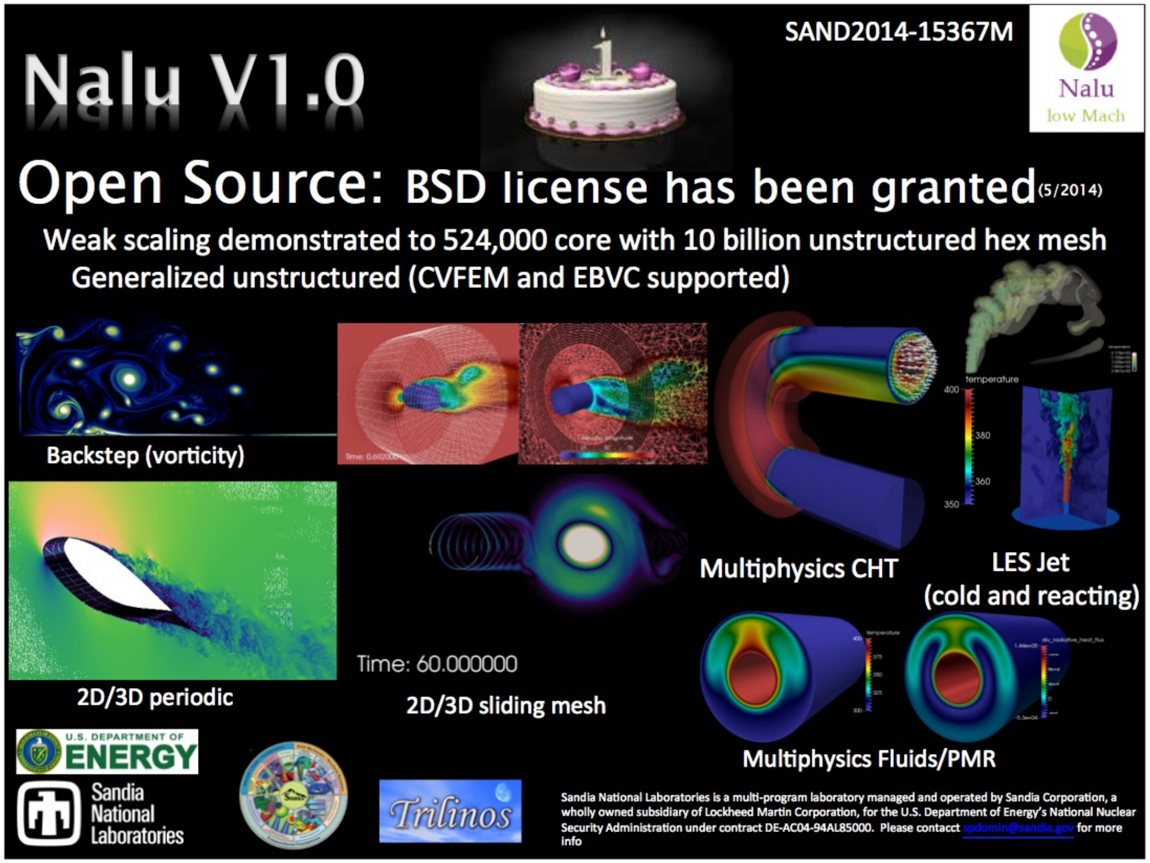
- You will receive an invitation to join (SUID)
- We are also working to provide linux-based laptops with a Nalu installation to facilitate our laboratory sessions on Fridays (one laptop per 2-3 students)

Visualization: Paraview (open source)

Grid Generation: We have created all of the meshes that you will need for this class. If you have ideas on specific problems that you would like to run, which are tractable given the cluster resources, let us know

- If you are familiar with another simulation tool, this code base can also be incorporated in the course
- The objective is not to learn Nalu, but rather to have a simulation tool to evaluate and test the concepts provided in class

Nalu: https://github.com/NaluCFD/Nalu



Course Grading

Grading

- Two laboratory-based write-ups (20%), see: Nalu/reg_tests/test_files/laboratory
  - You can choose the laboratory writeups including:
    - 1d_quad4_adv_diff 2d_quad9_helium 3d_hex8_open_jet 2d_quad4_channel ~~2d_tri3_quad4_street~~ 3d_tet4_pipe 2d_quad9_couette 3d_hex8_dam_break
  - Format will be a brief report, i.e., ~5 pages and can follow the write-up approach (LaTex), or your document editor program of choice (provided as pdfs)
- Midterm (40%): ~10 page report
  - Focused on the 2d_tri3_quad4_street case
- Final (40%): ~10 page report

Late submissions are discouraged and allowed only under special circumstances

Labs and Projects: Designed to Remove the Meshing Requirement of CFD

Midterm Project: Discussed in *third* laboratory session

- Flow around a circular cylinder (2D vortex street), see:
  - Nalu/reg_tests/test_files/laboratory/2d_tri3_quad4_street/write_up/2d_tri3_quad4_street_laboratory.pdf
- Focuses on numerical accuracy and parametric sensitivities

Final Project: Focus on physics investigation; cases documented in the course reader

- Flow/Thermal analysis, see Nalu/examples (hula hoop)
- Rotating shapes, see Nalu/examples (shapes)
- If you have other ideas, let me know by week 6

Logistics

- Projects can be carried out as a team (2-3 members max)
- Reports no longer than 10 pages, however, should read like a manuscript, e.g., introduction, theory, numerical experiments, findings, conclusion
- You can also propose using an alternative CFD code base
- Please avoid AI-based usage!

Incoming Expectations

Familiarity with:

- Basic Unix commands
  - >ls, cd, pwd, mv, cp, rm, …
  - See, for example, http://www.ee.surrey.ac.uk/Teaching/Unix/
- Editing a file under a Unix environment
  - vi, emacs, etc.
- Numerical methods including:
  - Explicit/Implicit
  - Finite difference, finite volume, finite element, etc.
- Programing language can be useful
- Fluids mechanics and the underlying PDE set
- Reporting tool, e.g., LaTex, Word, etc.

$$\int \frac{\partial \bar{\rho}}{\partial t} dV + \int \bar{\rho}\tilde{u}_j n_j \, dS = 0$$

Given filter: $\overline{\phi(\boldsymbol{x},t)} \equiv \int_{-\infty}^{+\infty} \phi(\boldsymbol{x}',t) G(\boldsymbol{x}'-\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}'$

$$\int \frac{\partial \bar{\rho}\tilde{u}_i}{\partial t} \mathrm{d}V + \int \bar{\rho}\tilde{u}_i\tilde{u}_j n_j \mathrm{d}S = \int \tilde{\sigma}_{ij} n_j \mathrm{d}S - \int \tau_{ij}^{sgs} n_j \mathrm{d}S + \int (\bar{\rho} - \rho_\circ) \, g_i \mathrm{d}V$$

$$\int \frac{\partial \bar{\rho}\tilde{h}}{\partial t} \mathrm{d}V + \int \bar{\rho}\tilde{h}\tilde{u}_j n_j \mathrm{d}S = -\int \bar{q}_j n_j \mathrm{d}S - \int \tau_{h,j}^{sgs} n_j \mathrm{d}S - \int \frac{\partial \bar{q}_i^r}{\partial x_i} \mathrm{d}V + \int \left(\frac{\partial \bar{P}}{\partial t} + \tilde{u}_j \frac{\partial \bar{P}}{\partial x_j} \right) \mathrm{d}V + \int \overline{\tau_{ij} \frac{\partial u_i}{\partial x_j}} \mathrm{d}V$$

$$\int \frac{\partial \bar{\rho}\tilde{Z}}{\partial t} \mathrm{d}V + \int \bar{\rho}\tilde{u}_j\tilde{Z} n_j \mathrm{d}S = -\int \tau_{Z,j}^{sgs} n_j \mathrm{d}S + \int \bar{\rho} D \frac{\partial \tilde{Z}}{\partial x_j} n_j \mathrm{d}S$$

Subgrid Stress (SGS):

$$\tau_{ij}^{sgs} \equiv \bar{\rho}(\widetilde{u_i u_j} - \tilde{u}_i\tilde{u}_j)$$
$$\tau_{h,j}^{sgs} \equiv \bar{\rho}\left(\widetilde{h u_j} - \tilde{h}\tilde{u}_j\right)$$
$$\tau_{Z,j}^{sgs} \equiv \bar{\rho}\left(\widetilde{Z u_j} - \tilde{Z}\tilde{u}_j\right)$$

Closure:

$$\tau_{ij}^{sgs} - \frac{1}{3}\delta_{ij}\tau_{kk}^{sgs} = -2\mu^t S_{ij}^*$$
$$\tau_{h,j}^{sgs} = -\frac{\mu_t}{\mathrm{Pr}_t}\frac{\partial \tilde{h}}{\partial x_j}$$
$$\tau_{Z,j}^{sgs} = -\bar{\rho} D_t \frac{\partial Z}{\partial x_j}$$

LES Models:

Smagorinsky: $\mu_t = \rho \left(C_s \Delta\right)^2 |\tilde{S}|$

k_{SGS}: $\mu_t = C_{\mu_\epsilon} \Delta k^{\mathrm{sgs}\,\frac{1}{2}}$

WALE: $\mu_t = \rho \left(C_w \Delta\right)^2 \dfrac{\left(S_{ij}^d S_{ij}^d\right)^{3/2}}{\left(S_{ij} S_{ij}\right)^{5/2} + \left(S_{ij}^d S_{ij}^d\right)^{5/4}}$

- Mixture fraction-based unsteady flamelet combustion with generalized heat loss (multi-D table look-up; presumed PDF)
- Soot: Koo et al. (2018) from Moss and Askit (2007)

- ME300C or similar (fundaments of numerical methods)
- ME351a/b or similar (fluid mechanics)

Questions?

Useful Abstractions

Real World

- The intended application

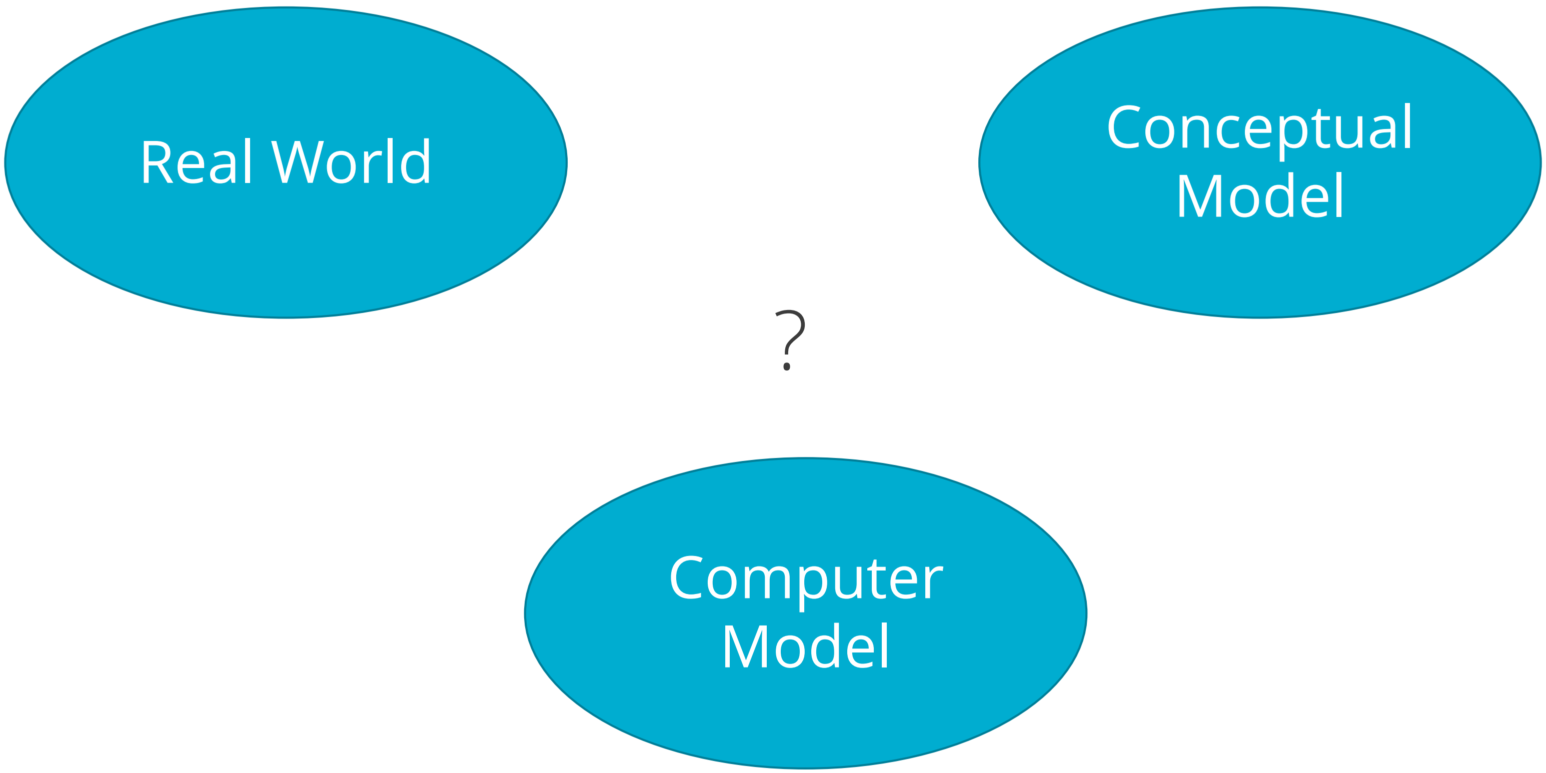
Conceptual Model

- Physical laws, hypothesis and models (e.g. a set of PDEs) and initial and boundary conditions

Computer Model

- A set of computational algorithms that allow to build a numerical, or approximated solution to the conceptual model

Mappings:



Qualification: ***Adequacy of the conceptual model to provide an acceptable level of agreement for the intended application***

Real World

Conceptual Model

Computer Model

Examples:

- Is the fluid Newtonian?
- Are compressibility effects important?
- Uniform or variable-density?
- Isothermal or non-isothermal?

Verification: ***Process of determining that (1) the model implementation faithfully represents the conceptual model (without mistakes) and (2) the solution to the model is accurate***

Real World

Conceptual Model

Verification

Computer Model

Examples:

- Is it RHS += A, RHS -=A? Which is correct?
- Should the contribution be scaled by a volume?
- How does the Quantity of Interest (QoI) behave as a function of mesh or time step resolution?

Validation: ***Process of determining the degree to which the numerical solutions (hence the model) is an accurate representation of reality from the perspective of the intended application***

Real World

Conceptual Model

Validation

Computer Model

Examples:

- Midterm project: compare drag coefficient prediction to an experiment
- Final project: compare drag coefficient for flow past rotating shapes; chaotic flow prediction at a given configuration

Verification vs Validation (V&V)? The Formal Lexicon

Verification: Are we solving the equations correctly?

- Represents an exercise in computational mathematics
- Given an equation, is the solution converging at known rates?

Validation: Are we solving the correct equations?

- Represents an exercise in understanding the physics associated with the real world use case

In this course, we will strongly focus on *verification*

- Establishing the correctness of the numerical implementation is key
- Comparisons of the numerical results to reality is not the primary objective

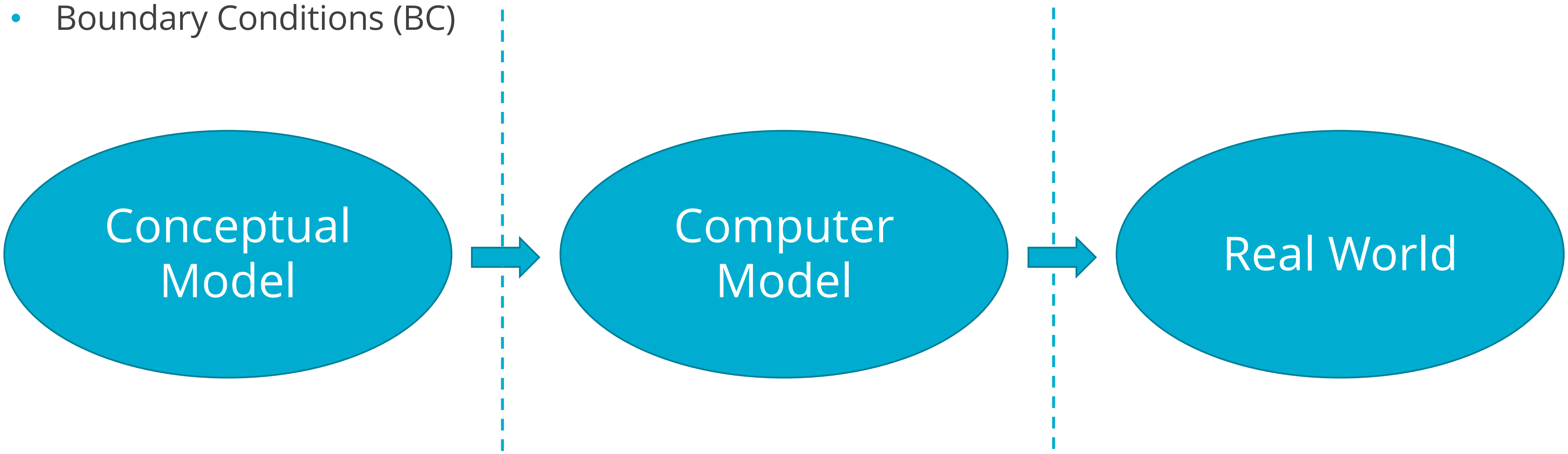
Verification challenges?

- Knowledge of the true solution, i.e., exact analytical solutions
- How many exact solutions exist for our class of physics? Not many!
- Hint: Method of Manufactured Solutions (MMS)

The Construction of a Numerical Method: Discretization and Solution

Mathematical Formulation

- Partial differential equations (PDE)
- Initial Conditions (IC)
- Boundary Conditions (BC)



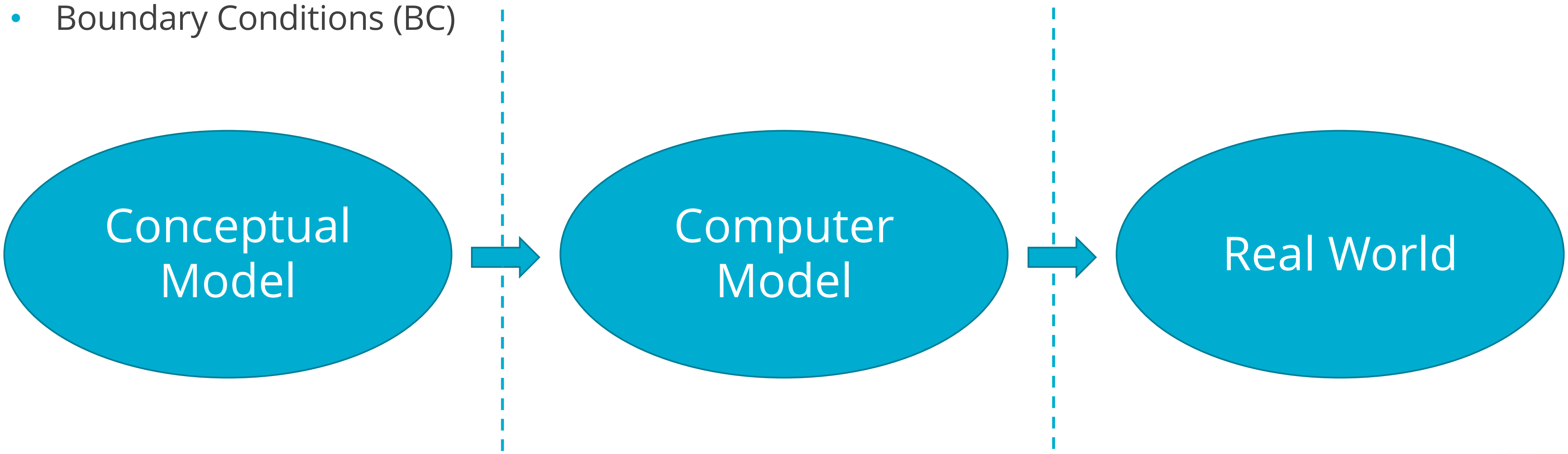
The Construction of a Numerical Method: Discretization and Solution

Mathematical Formulation

- Partial differential equations (PDE)
- Initial Conditions (IC)
- Boundary Conditions (BC)

Numerical Formulation

- Discretize in space and time
- Algebraic Differential Equations (ADE)



Discretization step

The Construction of a Numerical Method: Discretization and Solution

Mathematical Formulation

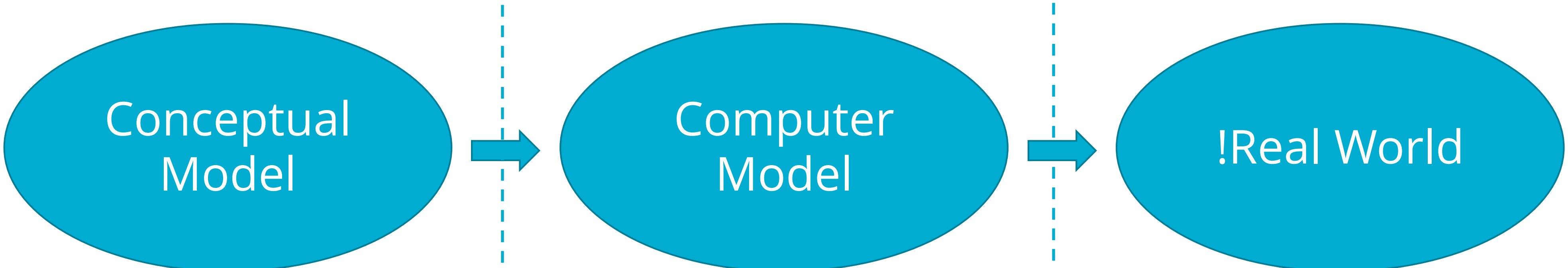
- Partial differential equations (PDE)
- Initial Conditions (IC)
- Boundary Conditions (BC)

Numerical Formulation

- Discretize in space and time
- Algebraic Differential Equations (ADE)

Numerical Solution

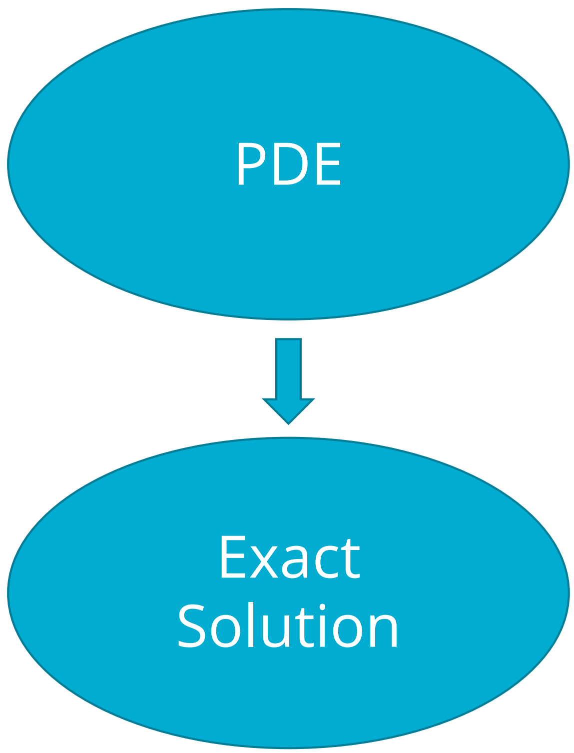
- Algebraic Differential Equations (ADE)
- Linear solvers
- Convergence, mesh adequacy, ….?



Discretization step

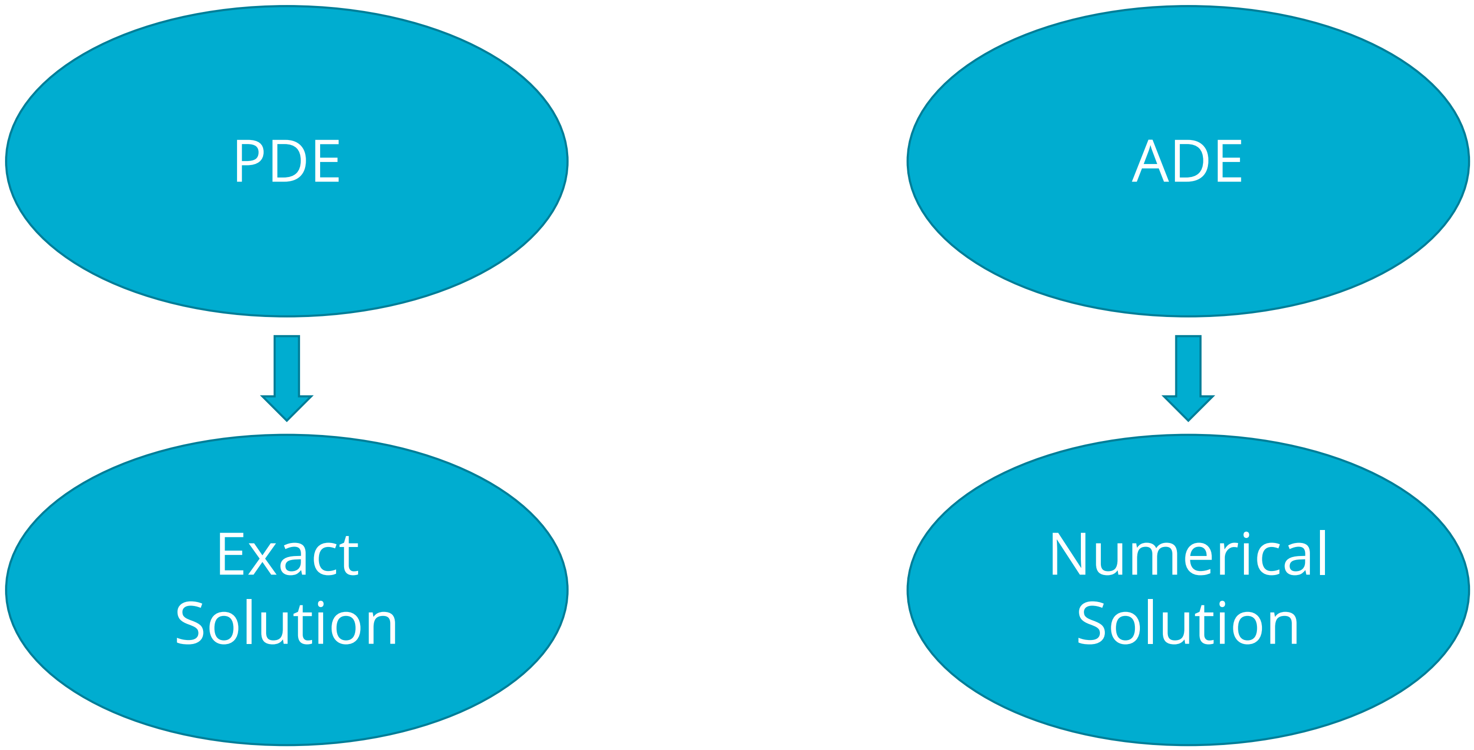
Solution step

Numerical Simulation Conundrum



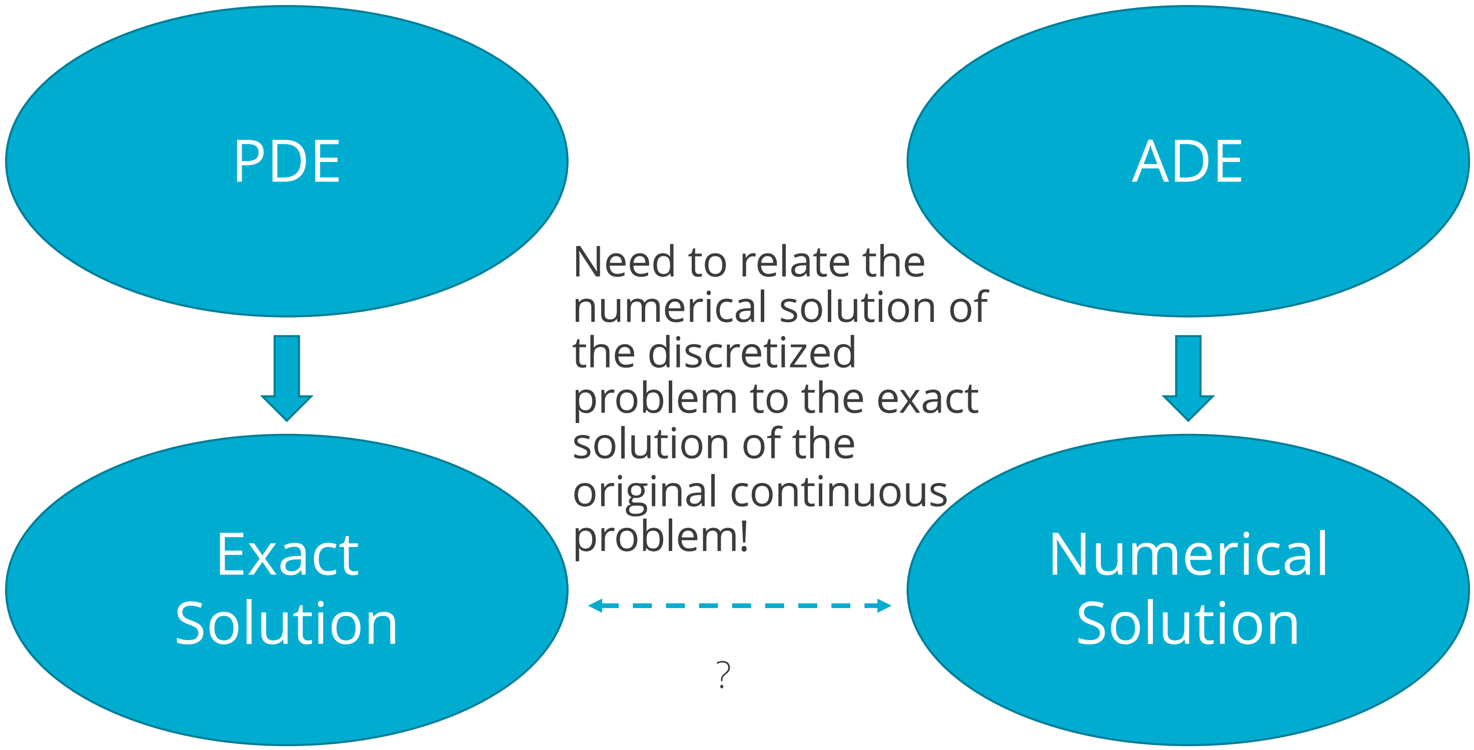
What we would like to have…

Numerical Simulation Conundrum



What we would like to have… What we have….

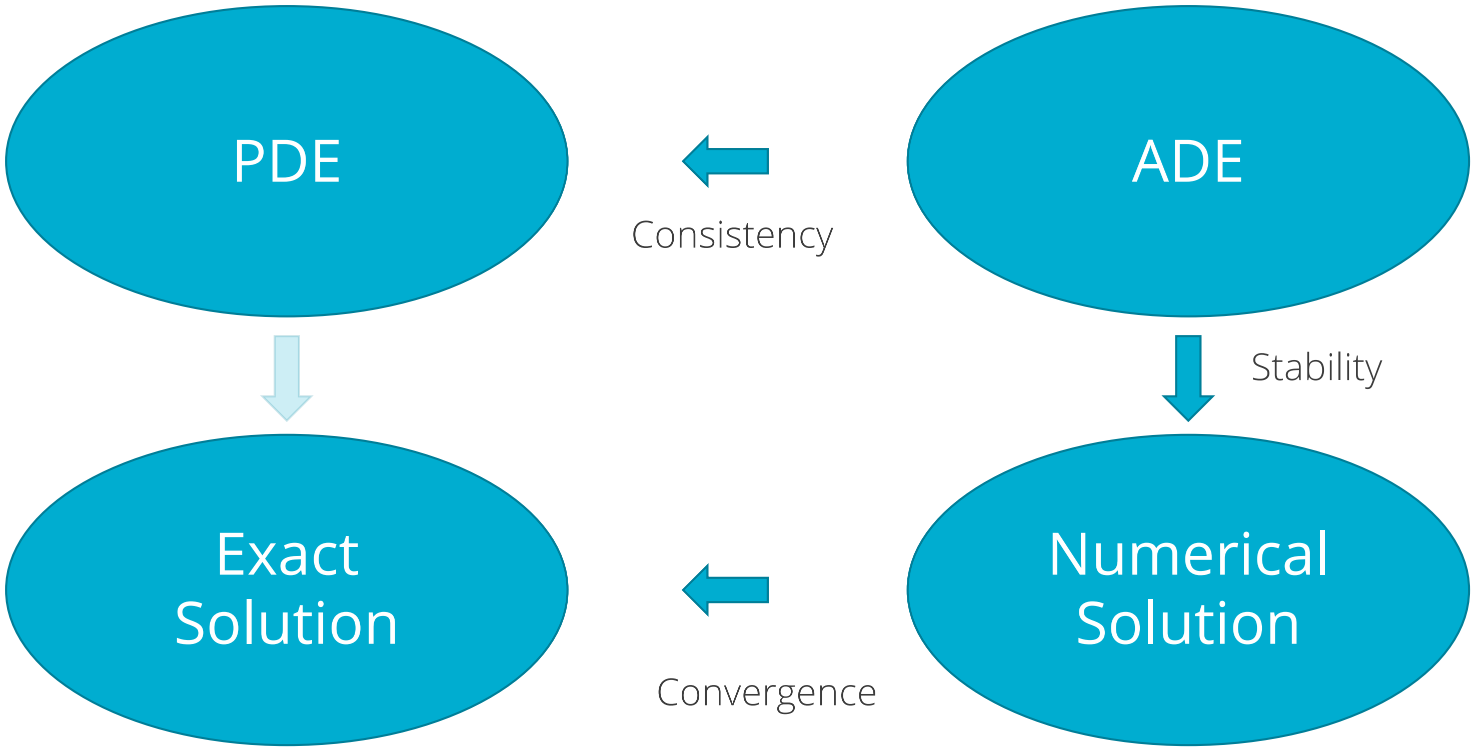
Numerical Simulation Conundrum



What we would like to have…

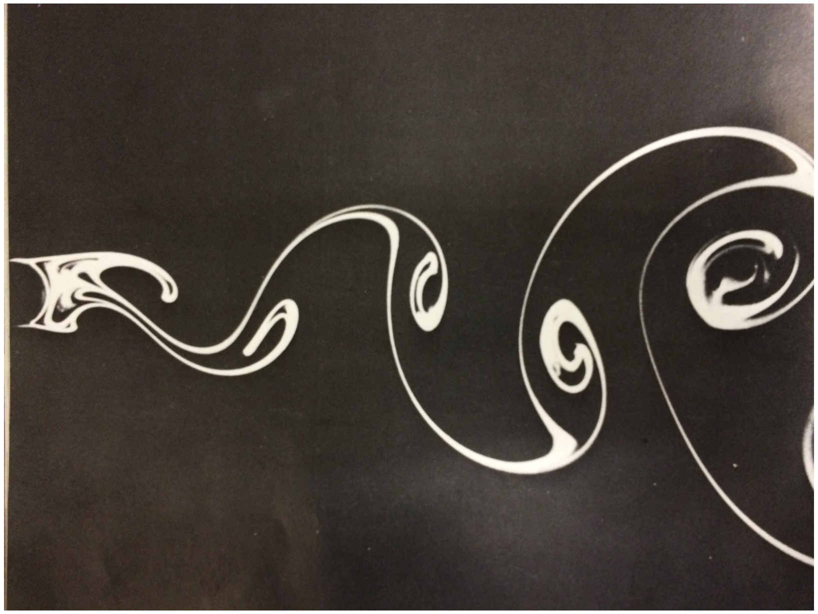
What we have….

Numerical Simulation Conundrum



- For a well-posed, linear PDE, Lax-Richtmyer Equivalence Theorem, AKA the *Fundamental Theorem of Numerical Analysis*: Consistency+Stability = Convergence

Example 1: Vortex Street



Conceptual Model

Computer Model

Real World?

Example 2: Modeling a Luminaria – In the Snow!

Conceptual Model

Computer Model

Real World?